

الفصل التاسع عشر

تحويل تطبيقات Access إلى مشروعات البيانات

يمكنك تحويل تطبيق Access إلى مشروع بيانات يعمل على قاعدة بيانات SQL Server من خلال معالج يسمى بمعالج التحويل Upsizing Wizard، حيث يقوم المعالج نيابةً عنك بتحويل التطبيق من خلال عدد من الشاشات التفاعلية المتتالية، إلا أن مشروع البيانات الناتج عادةً ما يحتوي على بعض القصور والمشاكل التي يجب حلها قبل استخدام مشروع البيانات المحول. سنتعرف في هذا الفصل على مفهوم التحويل وكيفية استخدام المعالج وكذلك كيفية علاج الأخطاء الناتجة من عملية التحويل.

بانتهاء هذا الفصل سنتعرف على:

- نظرة عامة على عملية التحويل Upsizing
- التجهيز لتحويل تطبيقات Access
- التحويل باستخدام طريقة المحاولة والخطأ Trial-and-Error
- تصحيح الأخطاء الناتجة عن معالج التحويل

نظرة عامة على عملية التحويل

تعتبر عملية تحويل أحد تطبيقات Access إلى قاعدة بيانات خلفية بتنسيق SQL Server 2005 أو SQL Server Express (SSX) ومشروع بيانات Access أو ما يسمى (ADP) Access Data Project من العمليات المركبة، حيث يقوم معالج التحويل Upsizing Wizard بإقصاء جميع كائنات البيانات الموجودة داخل Access والمثلة في الجداول والاستعلامات من التطبيق. ويعتمد نجاح وسرعة عملية التحويل بالطبع على حجم قاعدة بيانات Access التي ترغب في تحويلها ودرجة تعقيدها. فإذا أخذت في الاعتبار عند تصميمك لتطبيق Access أنك ستقوم بتحويله فيما بعد إلى مشروع بيانات، فلاشك بأن احتمال نجاح عملية التحويل سيزداد بشكل كبير. وتتمثل الخطوات التي يقوم بها معالج التحويل عند تحويل تطبيق Access إلى مشروع بيانات فيما يلي:

١. الدخول إلى الحاسب برقم حساب ينتمي إلى مجموعة مديري الحاسب Administrators إذا كان خادم SQL على نفس الحاسب وإلا قم باستخدام حساب موجود على المجال الخاص بالشبكة.
٢. إنشاء قاعدة بيانات SQL Server جديدة تحتوي افتراضياً على نفس اسم قاعدة البيانات الأصلية بالإضافة إلى الخاتمة SQL.
٣. نسخ تركيب جداول Access إلى جداول جديدة بقاعدة البيانات ثم إضافة البيانات إلى هذه الجداول.
٤. إضافة الفهارس إلى الجداول.
٥. إضافة الخصائص الإضافية إلى الجداول ثم تعيين قيم هذه الخصائص.
٦. إنشاء عرض داخل مجموعة الاستعلامات بلوحة التنقل لكل استعلام داخل قاعدة بيانات Access لا يحتوي على الجزء GROUP BY.
٧. إنشاء دالة سطرية (TVF) Inline table-valued function لكل استعلام

- داخل قاعدة بيانات Access يحتوي الجزء **GROUP BY**.
٨. إنشاء دالة سطرية لكل استعمال يحتوي على معاملات داخل قاعدة بيانات Access.
٩. إنشاء إجراء مخزن **Stored Procedure** لكل استعمال من الاستعلامات الإجرائية الموجودة بقاعدة بيانات Access وهي استعلامات الإضافة والتحديث والحذف وإنشاء جدول جديد.
١٠. إنشاء مشروع جديد بالامتداد **adp**. يحتوي على اتصال بقاعدة بيانات **SQL Server** الجديدة.
١١. نسخ جميع النماذج والنماذج الفرعية والتقارير والتقارير الفرعية ووحدات الماكرو والوحدات النمطية الموجودة داخل قاعدة البيانات Access ذات الامتداد **accdb**. إلى المشروع الجديد ذى الامتداد **adp**. ولا يقوم المعالج بإحداث أى تغييرات بكود **VBA** المستخدم داخل النماذج والتقارير أو الوحدات النمطية.

التجهيز لتحويل تطبيقات Access

- لزيادة إمكانية الحصول على عملية تحويل ناجحة، يمكنك استخدام طريقتين أساسيتين:
- طريقة المحاولة والخطأ أو **Trial and error** وفيها يتم تشغيل المعالج مكرراً بصورة متكررة لتحديد الجهد الذى يجب بذله لتحويل تطبيق Access بصورة صحيحة. فإذا كان بقاعدة البيانات جداول كبيرة الحجم، يمكنك استخدام الخيار **Structure-only** لتقليل الوقت المستغرق فى عملية التحويل. قم بعد ذلك بقراءة التقرير الناتج من المعالج وقم بإجراء التعديلات المطلوبة لتضمين الكائنات التى فشل المعالج فى تحويلها ثم حاول مرةً أخرى.
 - طريقة التحويل المخطط أو **Planned migration** وفيها يتم إجراء التغييرات على الجداول والاستعلامات والكائنات الأخرى الموجودة بتطبيق Access لتهيئتها لعملية التحويل مما يعمل على تقليل الأخطاء الناتجة من معالج التحويل فيما

بعد. فعلى سبيل المثال، يمكنك تحويل كود VBA الذى يستخدم كائنات Data Objects (DAO) Access إلى كائنات ActiveX أو ActiveX Data Objects (ADO) التى تستخدم بدورها مزود بيانات OLE DB.

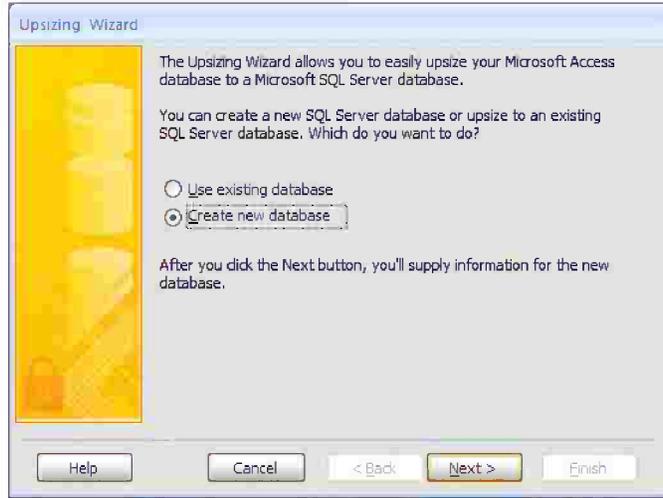
فإذا كان تطبيق Access صغيراً أو فى مراحله الأولى ولا يحتوى على كود VBA يشير إلى كائنات DAO، يمكنك استخدام الطريقة الأولى، طريقة المحاولة والخطأ. إلا أنه يفضل استخدام الطريقتين معاً للحصول على أفضل النتائج بعد عملية التحويل.

التحويل باستخدام طريقة المحاولة والخطأ Trial-and-Error

لكى نتعرف على ما يحدث عند تحويل تطبيقات Access إلى مشروعات بيانات من خلال معالج التحويل Upsizing Wizard، دعنا نقوم بتحويل قاعدة البيانات SalesProject.accdb ونرى ما يحدث. تابع معنا الخطوات الآتية:

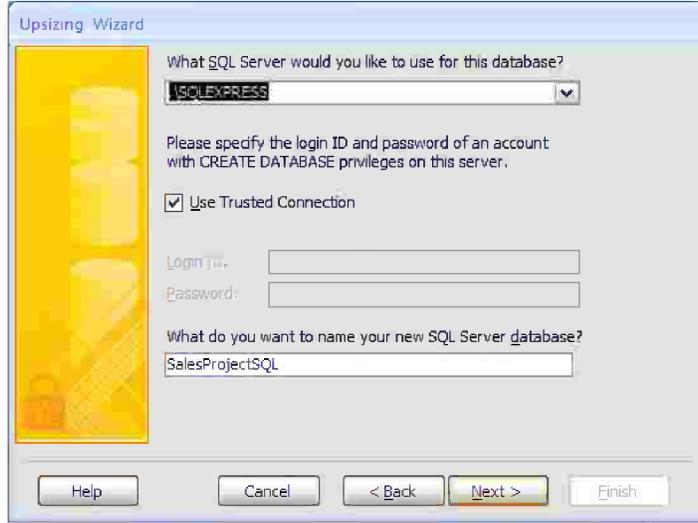
1. قم بإنشاء نسخة احتياطية من قاعدة البيانات التى ترغب فى تحويلها حتى يسهل الرجوع إليها فى أى وقت، وبالرغم من عدم تأثير معالج التحويل فى قاعدة البيانات الأصلية. ويمكنك من خلال المعالج تحويل ملف قاعدة بيانات واحد أو تحويل تطبيق مقسم إلى قاعدة بيانات أمامية وأخرى خلفية.
2. تأكد من عدم تشفير قاعدة البيانات التى ترغب فى تحويلها وإلا قم بإلغاء التشفير.
3. فى حالة وجود أى جداول أو استعلامات مخفية داخل قاعدة البيانات، انقر الكائن بزر الفأرة الأيمن داخل لوحة التنقل ثم اختر Properties لإظهار مربع الخصائص المصاحب للكائن ونشط التبويب General ثم قم بتعطيل مربع الاختيار Hidden Attributes.
4. سنقوم فى هذا المثال بتحويل قاعدة البيانات SalesProject.accdb الموجودة على القرص المدمج المرفق بالكتاب وهى نسخة من قاعدة البيانات Sales.accdb. قم بفتح قاعدة البيانات داخل Access ثم نشط التبويب Database Tools ثم انقر الزر  بالقسم Move Data

لبدء معالج التحويل، وحينئذٍ تظهر الشاشة الأولى من شاشات المعالج والتي تسألك إذا كنت ترغب في استخدام نفس قاعدة البيانات أم إنشاء قاعدة بيانات جديدة (انظر شكل ١٩-١).



شكل ١٩-١ الشاشة الأولى من معالج تحويل قاعدة البيانات.

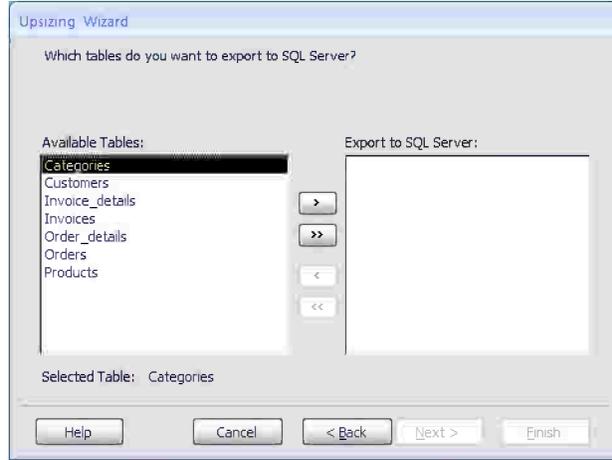
٥. ابق على الاختيار الافتراضي كما هو لأننا نرغب في إنشاء قاعدة بيانات جديدة ثم انقر زر **Next** للانتقال إلى الشاشة التالية والتي يتم من خلالها تحديد خادم **SQL** الذي سيحتوى على قاعدة البيانات وكذلك اسم قاعدة البيانات الجديدة (انظر شكل ١٩-٢).



شكل ٢-١٩ تحديد اسم الخادم واسم قاعدة البيانات الجديدة.

٦. من مربع التحرير والسرد الموجود بالجزء العلوى من الشاشة، اختر الحاسب الذى يحتوى على خادم SQL واسم الخادم أو قم بإدخال هذه البيانات بنفسك. فإذا كنت تستخدم SQL Server Express (SSX)، استخدم الاسم **SQLEXPRESS** هكذا **.\SQLEXPRESS**.
٧. فى حالة الدخول كأحد مستخدمى Windows الذين لهم صلاحيات الإدارة، نشط مربع الاختيار **Use Trusted Connection**، وإلا قم بإدخال **sa** أو المعرف الخاص بك إلى مربع النص **Login ID** وإدخال كلمة المرور إلى مربع النص **Password**.
٨. يقوم المعالج بتسمية قاعدة البيانات الجديدة بنفس اسم قاعدة البيانات الأصلية مضافاً إليه الخاتمة **SQL**. قم إن أحببت بتغيير الاسم من خلال مربع النص الموجود بالجزء السفلى من الشاشة.
٩. انقر زر **Next** للانتقال إلى الشاشة التالية من المعالج التى يمكنك من خلالها تحديد الجداول التى ترغب فى نسخها إلى قاعدة بيانات **SQL Server** الجديدة (انظر

شكل ٣-١٩).



شكل ٣-١٩ تحديد الجداول التي ترغب في تصديرها لقاعدة البيانات الجديدة.

١٠. انقر الزر **>>** لنقل جميع الجداول الموجودة بقاعدة بيانات Access من مربع

السرد **Available Tables** إلى مربع السرد **Export to SQL Server**.

١١. انقر زر **Next** للانتقال إلى الشاشة التالية من المعالج والتي يمكنك من خلالها تحديد

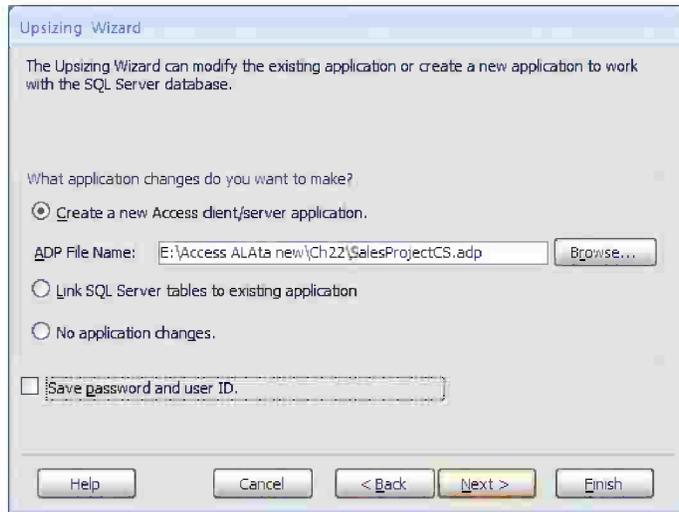
خصائص الجداول التي ترغب في تصديرها مع البيانات (انظر شكل ٤-١٩).



شكل ٤-١٩ تحديد خصائص الجداول التي ترغب في تصديرها مع البيانات.

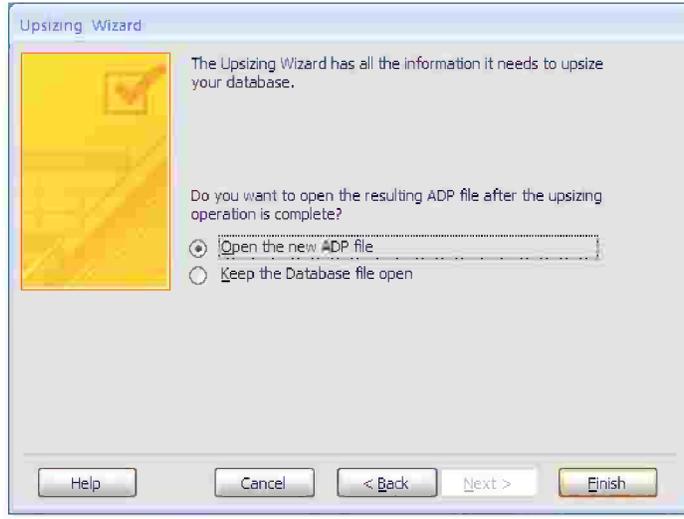
١٢. ابق على الخيارات الافتراضية كما هي والتي يتم فيها تصدير الفهارس وقواعد التحقق من الصحة والقيم الافتراضية والعلاقات بين الجداول مع استخدام التكامل المرجعي للبيانات. وفي حالة الجداول الكبيرة التي تحتوي على أكثر من ٢٥٠ ألف سجل، من الأفضل تنشيط مربع الاختيار **Only create the table structure; don't upsize any data** حتى يتم تصدير الجداول فقط مع عدم تصدير ما بها من بيانات.

١٣. انقر زر **Next** للانتقال إلى الشاشة التالية من المعالج والتي تخريك بين إنشاء مشروع بيانات جديد أو ربط جداول **SQL Server** بأحد تطبيقات **Access** (انظر شكل ٥-١٩). تأكد من تنشيط زر الاختيار **Create a new Access client/server application** لإنشاء مشروع بيانات جديد، حيث يقوم المعالج بتخصيص نفس اسم قاعدة بيانات **Access** مضافاً إليها الحاتمة **CS** ولكن مع استخدام الامتداد **.adp**.



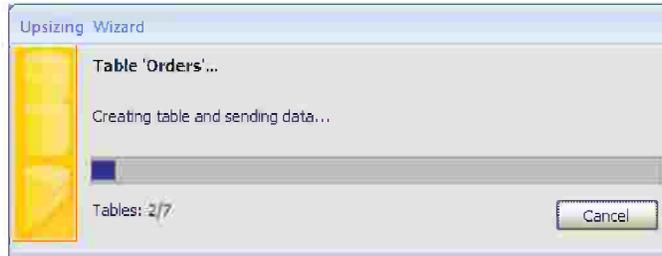
شكل ٥-١٩ يخريك المعالج بين إنشاء مشروع بيانات جديد أو استخدام مشروع موجود مسبقاً.
١٤. انقر زر **Next** للانتقال إلى الشاشة الأخيرة من المعالج والتي تسألك إذا كنت ترغب في فتح مشروع البيانات أم الإبقاء على قاعدة البيانات المحولة بمجرد انتهاء

المعالج من عملية التحويل (انظر شكل ١٩-٦). تأكد من تنشيط زر الاختيار **Open the new ADP file** لفتح مشروع البيانات.



شكل ١٩-٦ الشاشة الأخيرة من شاشات معالج التحويل.

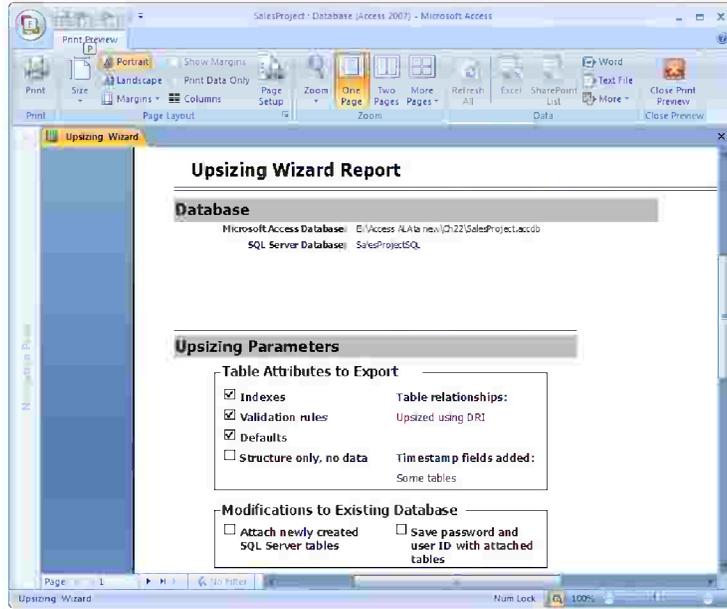
١٥. انقر زر **Finish** لإنهاء المعالج، وحينئذ يبدأ المعالج في عملية التحويل حيث يظهر شريط التقدم الذي يعبر عن مدى إتمام العملية (انظر شكل ١٩-٧)، كما قد تظهر بعض رسائل الخطأ أثناء تصدير الاستعلامات والناجئة عن عدم قدرة المعالج على التعامل مع تعبيرات **VBA** الموجودة داخل كود **SQL**. انقر زر **Ok** بعد قراءة كل رسالة لإغلاقها وإتمام عملية التحويل.



شكل ١٩-٧ يظهر المعالج مدى إنجاز عملية التحويل.

وبمجرد انتهاء المعالج من نسخ النماذج والكائنات الأخرى الموجودة بالتطبيق، والتي

تستغرق الكثير من الوقت مقارنةً ببقية الكائنات (ما عدا الجداول كبيرة الحجم)، يظهر تقرير التحويل بنافذة معاينة الطباعة (انظر شكل ٨-١٩). قم بحفظ التقرير إن أردت ثم قم بإغلاق نافذة المعاينة.



شكل ٨-١٩ يقوم المعالج بإظهار رسائل الخطأ أثناء تحويل بعض الاستعلامات.

وبالنظر إلى محتويات التقرير الناتج تجد أنه لا يوجد مشاكل عن عملية التحويل، ولعل السبب في ذلك أن قاعدة البيانات التي بين أيدينا صغيرة ولا تحتوي علي استعلامات معقدة، ربما نجد بعض القيود والعيوب الناتجة من معالج التحويل وهي:

- لا يقوم المعالج بحذف كلمة **DISTINCTROW** (وهي إحدى الكلمات الأساسية المستخدمة داخل **SQL** فقط) كما أنه لم يتم بتحويلها إلى كلمة **DISTINCT** المستخدمة داخل **T-SQL**.
- لا يقوم المعالج بتحويل الاستعلامات المتداخلة **Crosstab Queries** وذلك لأن **SQL Server 2005** لا يدعم الكلمة الأساسية **TRANSFORM** المستخدمة داخل **SQL** كما أنه لا يستخدم نفس صيغة الكلمة الأساسية **PIVOT**.

- لا يستطيع المعالج التعامل مع كود VBA وذلك لعدم احتواء T-SQL على مكافئ للكلمات الأساسية الموجودة داخل VBA.
- في حالة عدم قدرة المعالج على تحويل الاستعلام الرئيسي (الخارجي)، فلن يتمكن أيضاً من تحويل الاستعلامات الداخلية لهذا الاستعلام.
- لا يستطيع المعالج تحويل استعلامات التحديد والاستعلامات الإجرائية التي تحتوي على عبارة IN المستخدمة في الإشارة إلى قاعدة بيانات أخرى.
- لا يستطيع المعالج تحويل الاستعلامات الإجرائية التي تحتوي على معاملات، بينما يتم تحويل استعلامات التحديد التي تحتوي على معاملات إلى دوال سطرية TVF ذات معاملات.
- لا يستطيع المعالج تحويل الاستعلامات التي تحتوي على كلمات محجوزة داخل SQL Server كمسميات للأعمدة مثل OrderDate AS Date.
- لا يستطيع المعالج تحويل الاستعلامات الغير قابلة للتنفيذ نتيجة خطأ في اسم أحد جداول الاستعلام أو الحقول المستخدمة بداخله.

تشغيل الجولة الثانية من معالج التحويل

بعد إجراء التعديلات المناسبة على الاستعلامات بالاستعانة بالنقاط التي ذكرناها منذ قليل وكما سنوضح بالجزء المتبقى من هذا الفصل، يجب تشغيل معالج التحويل مرةً أخرى. وذلك باتباع الخطوات الآتية:

1. قم بإغلاق جميع كائنات المشروع adp. الذي قمنا بإنشائه منذ قليل إذا كانت مفتوحة بالفعل ما عدا إطار قاعدة البيانات.
2. انقر الزر  بالركن الأيسر العلوي من الشاشة ثم اختر Server من القائمة المنسدلة الناتجة ومنها اختر SQL Database Drop ثم انقر زر Yes رداً على مربع الرسالة الذي يختبر جديتك في فصل قاعدة بيانات SQL Server (وهي قاعدة البيانات SalesProjectSQL.adp في هذه الحالة) عن التطبيق ولاحظ

ظهور قائمتى الجداول والاستعلامات خاليتين.

٣. قم بإغلاق المشروع ثم استخدم مستكشف Windows في حذفه (الملف SalesProjectCS.adp في هذه الحالة).

٤. قم بفتح قاعدة البيانات التي ترغب في تحويلها (وهي SalesProject.accdb في هذه الحالة بعد إصلاح ما بها من أخطاء) ثم قم بتكرار الخطوات من ٤ إلى ١٥ المذكورة بعملية التحويل الأولى.

بمجرد انتهاء المعالج من عملية التحويل، تختفي جميع الأخطاء التي كانت موجودة في الحالة الأولى ما عدا بعض الاستعلامات التي تحتوي على أخطاء، وبدلاً من تضييع الوقت في علاج هذه الأخطاء ثم إجراء عملية التحويل مرةً أخرى، قد يكون من الأفضل تصحيح هذه الاستعلامات داخل المشروع الناتج.

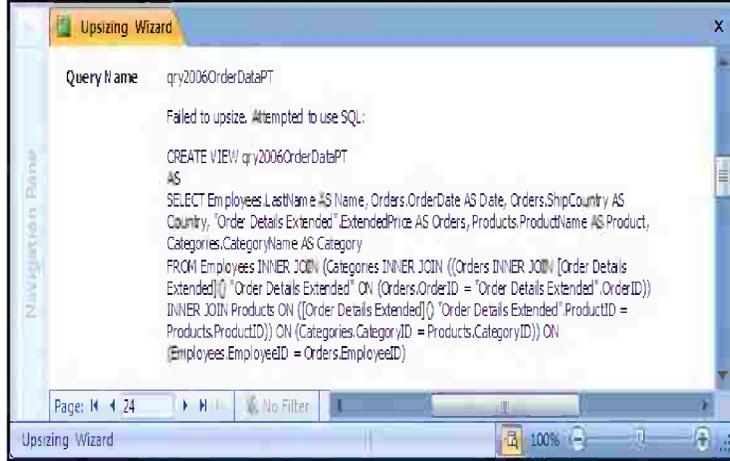
تصحيح أخطاء المعالج

على الرغم من عدم وجود أخطاء أثناء تحويل التطبيق Sales.accdb إلى مشروع بيانات، إلا أنك ربما صادفت أخطاءً عند تحويل تطبيقات أخرى. وقد ذكرنا منذ قليل الإجراءات التي يجب مراعاتها في قاعدة البيانات قبل تحويلها، وسنقوم فيما يلي بإلقاء الضوء على أشهر هذه الأخطاء وكيفية التعامل معها.

العمل مع الاستعلامات الداخلية

يوضح شكل ٩-١٩ التالي الجزء الخاص بأحد الاستعلامات داخل التقرير الناتج من معالج التحويل. على فرض أن لدينا استعلاماً داخلياً يستخدم أحد الاستعلامات الأخرى الموجودة داخل التطبيق؛ وليكن الاستعلام Order Details Extended، كمصدر لبياناته، حيث يستخدم هذا الاستعلام الداخلي كمصدر لجدول محوري PivotTable يقوم بعرض طلبات الصرف ربع الشهرية أو الشهرية مرتبة باسم الموظف والمجموعة.

الفصل التاسع عشر : تحويل تطبيقات Access إلى مشروعات البيانات



شكل ٩-١٩ الجزء الخاص بالاستعلام داخل تقرير التحويل.

ويحتوي كود SQL Access المصاحب للاستعلام على الكود التالي:

```
SELECT Employees.LastName AS Name,
Orders.OrderDate AS [Date],
Orders.ShipCountry AS Country,
[Order Details Extended].ExtendedPrice AS Orders,
Products.ProductName AS Product,
Categories.CategoryName AS Category
FROM Employees
INNER JOIN (Categories
INNER JOIN ((Orders
INNER JOIN [Order Details Extended]
ON Orders.OrderID =
[Order Details Extended].OrderID)
INNER JOIN Products
ON [Order Details Extended].ProductID =
Products.ProductID)
ON Categories.CategoryID = Products.CategoryID)
ON Employees.EmployeeID = Orders.EmployeeID;
```

وعند تحويل المعالج لهذا الاستعلام، يقوم بترجمة العبارة FROM بطريقة خاطئة فنحصل على الكود التالي:

```
FROM Employees
INNER JOIN (Categories
INNER JOIN ((Orders
```

```
INNER JOIN [Order Details Extended]()
"Order Details Extended"
ON (Orders.OrderID =
"Order Details Extended".OrderID))
INNER JOIN Products ON ([Order Details Extended]()
"Order Details Extended".ProductID =
Products.ProductID))
ON (Categories.CategoryID = Products.CategoryID))
ON (Employees.EmployeeID = Orders.EmployeeID)
```

والخطأ هنا حقيقةً في العبارة التي تقوم باستدعاء الدالة (`Order Details Extended()`). فإذا قمنا بحذف حالتنا استدعاء هذه الدالة من الكود السابق، يتم حل المشكلة من جذورها، وهذا على فرض تحويل الاستعلام `Orders Details Extended` إلى عرض `View` داخل المشروع الناتج. ولكن نظراً لعدم تحويل الاستعلام بنجاح، فلن تتمكن من تعديل كود `T-SQL` الناتج، وإنما يمكنك حل هذه المشكلة باستخدام أي من الطريقتين التاليتين:

- استبدال الدالة (`Order Details Extended()`) بعرض جديد. لأداء ذلك، قم بنسخ كود `T-SQL` الخاص بالدالة ثم قم بحذف الدالة ثم إنشاء عرض جديد داخل مصمم المشروعات وقم بلصق الكود بلوحة `SQL` الخاصة بالعرض الجديد. ونظراً لعدم تحويل الاستعلام، قم بنسخ كود `Access SQL` الخاص بالاستعلام ثم قم بإنشاء عرض جديد وقم بلصق الكود بلوحة `SQL` الخاصة بالعرض الجديد.
- قم بنسخ عبارة `Access SQL` الخاصة بالاستعلام الأساسي ثم قم بإنشاء عرض جديد داخل مصمم المشروعات وقم بلصق العبارة داخل لوحة `SQL` الخاصة بالعرض.

وتجدر الإشارة هنا إلى ضرورة اختبار كل عنصر على حده داخل مصمم المشروعات للتأكد من عمله بشكل صحيح.

العمل مع الحقول المحسوبة

يمكنك داخل Access SQL استخدام القيمة الناتجة من أحد الحقول (الأعمدة) المحسوبة كمصدر لحقل محسوب آخر، وهي الحالة التي يتم فيها تخزين قيم المبيعات أو الضريبة. فعلى سبيل المثال، يعمل الكود التالي بشكل سليم عند استخدام Access SQL، بينما لا يعمل مع T-SQL:

```
SELECT OrderTotal * 0.06 AS StateTax,  
       OrderTotal * 0.01 AS CountyTax,  
       OrderTotal * 0.005 AS CityTax,  
       StateTax + CountyTax +  
       CityTax AS Taxes,  
       OrderTotal + Taxes AS InvoiceTotal  
FROM Orders
```

إذا صادف معالج التحويل استعلاماً يحتوي على حقول محسوبة كما في الكود السابق، لا تتم عملية التحويل بشكل صحيح لأن كل عنصر داخل عبارة SELECT يجب أن يحتوي على مرجع صريح لأحد الحقول الموجودة بالجدول الموجود بالجزء FROM، وهذا ما لا يتحقق مع الحقل المحسوب Taxes في الكود السابق، لذا لا يقوم مفسر SQL Server بترجمة هذا الكود، وهذا ما يظهر جلياً مع الاستعلام التالي الذي يتم فيه حساب تكلفة الشحن كنسبة من قيمة كل طلب صرف، حيث تم حذف الدالة Format() المستخدمة في إظهار النسبة المئوية من الاستعلام قبل عملية التحويل، كما تم استبدال الدالة CCur() المستخدمة في تحويل القيمة من النوع Double إلى النوع Currency (وهي إحدى الدوال المعرفة داخل VBA) بالدالة CAST() وهي الدالة المصاحبة داخل T-SQL. يوضح الكود التالي عبارة SQL المصاحبة للاستعلام داخل Access:

```
SELECT Customers.CompanyName,  
       Customers.PostalCode,  
       Customers.Country,  
       Orders.OrderID, Orders.OrderDate,  
       Sum(CCur([UnitPrice]*[Quantity]*  
            (1-[Discount]))) AS Amount,  
       Orders.Freight, [Freight]/[Amount] AS FreightPct  
FROM (Customers
```

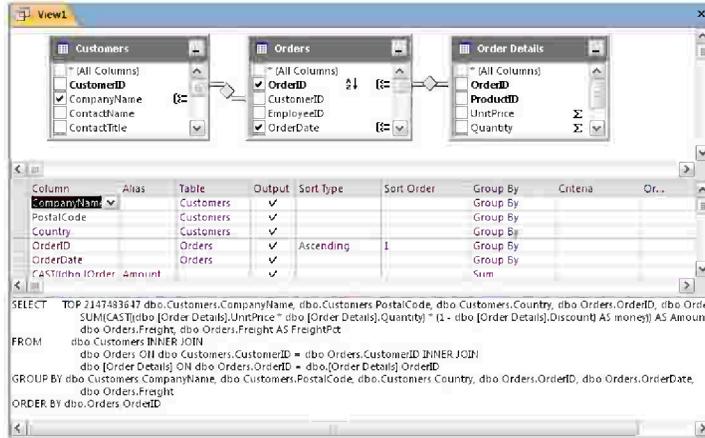
```
INNER JOIN Orders
ON Customers.CustomerID=Orders.CustomerID)
INNER JOIN [Order Details]
ON Orders.OrderID=[Order Details].OrderID
GROUP BY Customers.CompanyName, Customers.PostalCode,
Customers.Country, Orders.OrderID,
Orders.OrderDate, Orders.Freight
ORDER BY Orders.OrderID;
```

ويكمن التعارض في هذه العبارة في الحقل المحسوب **FrieghtPct** بالجزء
[Freight]/[Amount] AS FreightPct

ويمكنك حل هذه المشكلة بالاستعلام الأساسي داخل **Access** عن طريق استبدال
[Amount] بالجزء **Sum AS Amount** إلا أنك ستحتاج في هذه الحالة إلى تشغيل
معالج التحويل مرةً أخرى، لذا فالأفضل إنشاء عرض جديد داخل المشروع الناتج. لأداء
ذلك، تابع معنا الخطوات الآتية:

١. انسخ عبارة **Access SQL** الخاصة بالاستعلام.
٢. قم بإنشاء عرض جديد داخل نافذة مصمم المشروعات ثم انقر الزر  لإظهار لوحة **SQL** وقم ب لصق الكود الذى قمت بنسخه من قبل داخل اللوحة.
٣. استبدل دوال التحويل (وهى الدالة **CCur()** في هذه العبارة) بالدالة **CAST** أو **CONVERT** والتي تحتوى على الصيغة **CAST(expression as money)**.
٤. احذف الجزء **[Amount]** وهو الجزء الذى تسبب في عدم تحويل الاستعلام من خلال معالج التحويل.
٥. انقر زر  . يقوم مصمم المشروعات بترجمة كود **T-SQL** وإنشاء العرض كما يقوم بإضافة الجزء **TOP 100 PERCENT** إلى بداية العبارة **SELECT**. قم بتغيير هذا الجزء إلى **TOP 2147483647** (انظر شكل ١٠-١٩).

الفصل التاسع عشر : تحويل تطبيقات Access إلى مشروعات البيانات



شكل ١٠-١٩ العرض الجديد داخل مشروع البيانات.

٦. استبدال الاسم المستعار الذي قيمته بـ [Amount] بالتعبير المصاحب وهو:

$$\frac{\text{SUM}((\text{dbo}.[\text{Order Details}].\text{UnitPrice} * \text{dbo}.[\text{Order Details}].\text{Quantity}) * (1 - \text{dbo}.[\text{Order Details}].\text{Discount}))}{\text{SUM}((\text{dbo}.[\text{Order Details}].\text{UnitPrice} * \text{dbo}.[\text{Order Details}].\text{Quantity}) * (1 - \text{dbo}.[\text{Order Details}].\text{Discount}))}$$
 ولا ضرورة هنا لاستخدام الدالة CAST() لأن القيمة الناتجة من العمود عبارة عن كسر عشري وليست قيمة نقدية. وبذلك يصبح كود العرض كما يلي:

```
SELECT TOP 2147483647 dbo.Customers.CompanyName,
    dbo.Customers.PostalCode, dbo.Customers.Country,
    dbo.Orders.OrderID, dbo.Orders.OrderDate,
    SUM(CAST((dbo.[Order Details].UnitPrice *
    dbo.[Order Details].Quantity) *
    (1 - dbo.[Order Details].Discount) AS money)) AS Amount,
    dbo.Orders.Freight,
    dbo.Orders.Freight /
    SUM((dbo.[Order Details].UnitPrice *
    dbo.[Order Details].Quantity) *
    (1 - dbo.[Order Details].Discount)) AS FreightPct
FROM
    dbo.Customers
    INNER JOIN dbo.Orders
        ON dbo.Customers.CustomerID =
        dbo.Orders.CustomerID
    INNER JOIN dbo.[Order Details]
        ON dbo.Orders.OrderID =
        dbo.[Order Details].OrderID
```

```
GROUP BY dbo.Customers.CompanyName,  
         dbo.Customers.PostalCode, dbo.Customers.Country,  
         dbo.Orders.OrderID, dbo.Orders.OrderDate,  
         dbo.Orders.Freight  
ORDER BY dbo.Orders.OrderID
```

٧. انقر الزر  للتأكد من صحة التعديلات التي قمت بها ثم قم بحفظ العرض بنفس اسم الاستعلام الأصلي.
٨. انقر الزر  لتشغيل العرض وتأكد من الحصول على نفس النتيجة التي حصلت عليها عند تشغيل الاستعلام الأصلي. الفرق هنا هو ظهور قيم العمود **FreightPct** في صورة كسور عشرية بدلاً من نسبة مئوية.
٩. لتغيير تنسيق قيم العمود **FreightPct**، انقر بزر الفأرة الأيمن في أي مكان بمصمم المشروعات ثم اختر **Properties** من القائمة الموضعية الناتجة لإظهار مربع الخصائص. نشط التبويب **Columns** ثم اختر **FreightPct** من القائمة **Column Name** ثم اختر **Percent** من القائمة **Format** (انظر شكل ١١-١٩).
١٠. أغلق مربع الخصائص و قم بحفظ العرض وتشغيله مرةً أخرى.



شكل ١١-١٩ تغيير تنسيق العمود من مربع الخصائص.

العمل مع الدوال التي تشير إلى قيم كائنات أدوات التحكم

تقبل الاستعلامات المنشأة داخل Access القيم الناتجة من الكائنات المختلفة كمربع النص ومربع السرد والتحرير ومربع السرد وغيرها من أدوات التحكم الأخرى. وعند العمل مع الاستعلامات يوجد ثلاث قواعد أساسية يجب أخذها في الحسبان عند اختبار المشروعات المحولة وهي:

- عدم الثقة في المعالج، فعلى الرغم من معرفة عدم قدرة معالج التحويلات على تحويل الاستعلامات التي تشير بدورها إلى قيم بكائنات Access المختلفة، إلا أن المعالج يحاول أداء ذلك وبالتالي يقوم بإنشاء دوال وإجراءات مخزنة لا تعمل بشكل صحيح بقاعدة بيانات SQL Server الناتجة، لذا يجب أن تقوم بنفسك بمقارنة نتيجة تنفيذ كل استعلام محوّل بالاستعلام المصاحب داخل قاعدة بيانات Access.
- ابحث عن كل كائن مبني على استعلام محوّل وقم باختباره، ويمكنك في ذلك

الاستعانة بلوحة المهام **Object Dependencies** لإيجاد مراجع كل استعمال على حده.

- تأكد من أن الاستعلام المحوّل عبارة عن مصدر لنموذج أو تقرير محدد. فعلى فرض احتواء أحد الاستعلامات على عبارة **SQL** التالية (والتي تم فيها تغيير **DISTINCT** إلى **DISTINCTROW**):

```
SELECT DISTINCT Invoices.*
FROM Invoices
WHERE (((Invoices.OrderID) =
Forms!Orders![OrderID]));
```

فإن عبارة **T-SQL** المصاحبة للعرض الناتج من معالج التحويل تحتوى على الكود التالي:

```
SELECT DISTINCT dbo.Invoices.*
FROM dbo.Invoices
WHERE (OrderID = @Forms_Orders_OrderID)
```

وهنا قام المعالج باستبدال الرمزين ! و [] بالرمز _ .

اكتشاف الكائنات المعتمدة على استعلامات

أحياناً يكون من الصعب التعرف على طريقة توظيف أحد النماذج لأحد الاستعلامات وذلك لأن الاستعلام لا يظهر داخل الخاصية **Filter** الموجودة بمربع خصائص النموذج، وإنما تكمن العلاقة بين النموذج من ناحية والاستعلام من ناحية أخرى في وجود زر داخل النموذج يحتوى على حدث **VBA** يتم تنفيذه بمجرد نقر هذا الزر. ففي الكود التالي على سبيل المثال، والذي يحتوى على حدث نقر الزر **PrintInvoice** الموجود داخل النموذج **Orders** ويقوم بدوره بطباعة الفاتورة الخاصة بطلب الصرف الحالى داخل النموذج، يتم استخدام الاستعلام **invoice Filter** كعامل تصفية بالإجراء **OpenReport** هكذا:

```
Sub PrintInvoice_Click()
' This code created by Command Button Wizard.
On Error GoTo Err_PrintInvoice_Click
Dim strDocName As String
strDocName = "Invoice"
' Print Invoice report, using Invoices Filter query to print
' invoice for current order.
DoCmd.OpenReport strDocName, acViewNormal, "Invoices
```

Filter"

Exit_PrintInvoice_Click:
Exit Sub

Err_PrintInvoice_Click:
' If action was cancelled by the user, don't display an error message.
Const conErrDoCmdCancelled = 2501
If (Err = conErrDoCmdCancelled) Then
Resume Exit_PrintInvoice_Click
Else
MsgBox Err.Description
Resume Exit_PrintInvoice_Click
End If
End Sub

ففي هذا الكود يتسبب السطر التالي:

DoCmd.OpenReport strDocName,acViewNormal,"Invoices Filter"
في فتح التقرير Invoice مع تطبيق عامل التصفية Invoice Filter (الاستعلام) على هذا التقرير. فهنا مصدر بيانات التقرير عبارة عن الاستعلام Invoice Filter الذي قام معالج التحويل Upsizing Wizard بتحويله إلى عرض غير قابل للتنفيذ.

تحويل عبارات SQL المستخدمة داخل النماذج والتقارير وأدوات التحكم

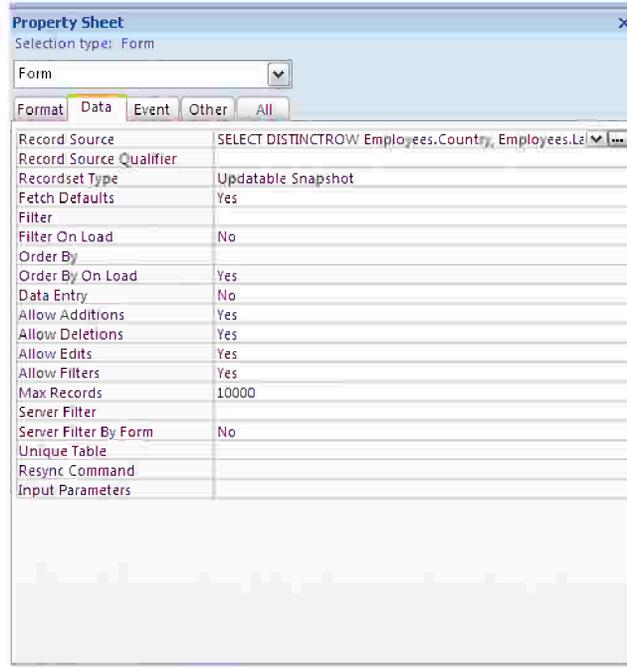
يستخدم العديد من مطوري Access عبارات SELECT في تعيين مصادر بيانات النماذج والتقارير وأدوات التحكم بدلاً من استخدام الاستعلامات المحفوظة بقاعدة البيانات. ولسوء الحظ أن هذا الكود لا يلتفت إليه معالج التحويل عند تحويل قاعدة البيانات إلى مشروع، لذا يجب أن تقوم بالتعامل مع هذا الكود بنفسك لتصحيح أي جزء بداخله لا يتفق مع SQL Server، حيث تكمن معظم الأخطاء في وجود المعرف

DISTINCTROW الغير مستخدم داخل SQL Server.

للتعرف على كيفية حل هذا النوع من المشاكل، تابع معنا الخطوات الآتية:

١. قم بإظهار النموذج أو التقرير بطريقة عرض التصميم.

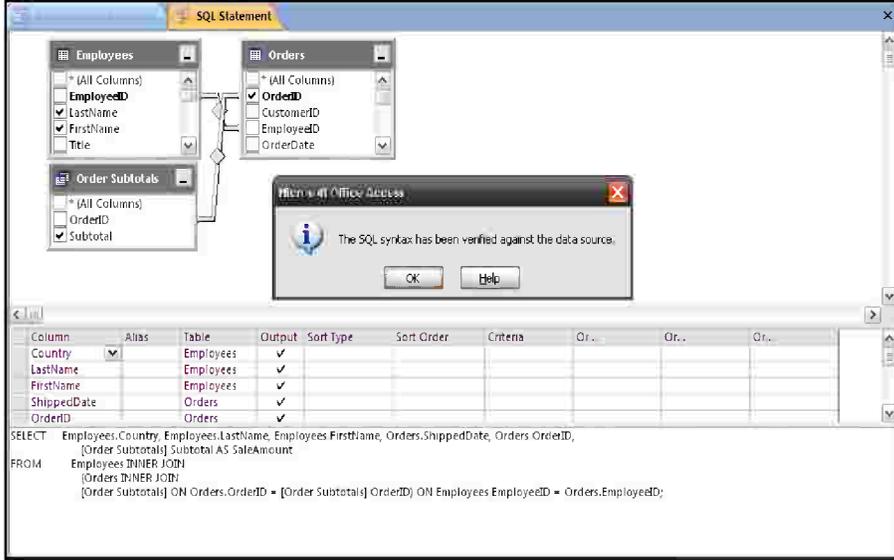
٢. قم بإظهار مربع الخصائص ثم نشط التبويب Data (انظر شكل ١٢-١٩).



شكل ١٢-١٩ تحتوي الخاصية Record Source على عبارة SQL بها كلمة DistinctRow. ٣. انتقل إلى محتويات الخاصية Record Source وقم بحذف كلمة DISTINCTROW من عبارة SQL ثم انقر الزر المجاور لإظهار نافذة العرض داخل نافذة مصمم المشروعات.

٤. قم بتعديل أى أخطاء موجودة بالكود، حيث يمكنك الاستعانة في ذلك بالزر . انقر هذا الزر لاختبار عبارة SQL وإظهار مربع رسالة يساعدك على الوصول إلى المشكلة إن وجدت. ولأن العبارة التي بين أيدينا لا تحتوي على أى أخطاء سوى استخدام كلمة DISTINCTROW التي قمنا بحذفها في الخطوة رقم ٢، نحصل على رسالة تفيد بصحة عبارة SQL (انظر شكل ١٣-١٩).

الفصل التاسع عشر : تحويل تطبيقات Access إلى مشروعات البيانات



شكل ١٣-١٩ تفيد الرسالة عدم وجود أى أخطاء بعبارة SQL بعد حذف كلمة DISTINCTROW منها

٥. أغلق نافذة مصمم المشروعات وانقر زر Yes رداً على مربع الرسالة الذى يسألك إذا كنت ترغب فى حفظ تعديلاتك. قم أيضاً بإغلاق النموذج وحفظ التعديلات التى تمت عليه.

٦. كرر الخطوتين ٢١ و ٢٠ مع النماذج الأخرى التى تحتوى على كلمة DISTINCTROW. بكوند SQL الموجود بالخاصية Record Source.

٧. قم بتشغيل النماذج للتأكد من عدم تأثير التغييرات التى قمت بها على البيانات الناتجة وتأكد من مطابقتها للبيانات التى نحصل عليها عند فتح هذه النماذج داخل تطبيق Access قبل إجراء عملية التحويل.

يمكنك تطبيق نفس الخطوات السابقة مع أدوات التحكم التى تحتوى قيم الخاصية Row Source فيها على عبارات SQL.

